****



**PLC Applications**

Module 3: Time sequence processes



PREPARED BY

**Academic Services Unit**

December 2011

© Applied Technology High Schools, 2011

Module 3: Time sequence processes

**Module Objectives**

Upon successful completion of this module, students will be able to:

* Analyze a control task that contains a time sequence processes.
* Understand and use ON/OFF timer, Asynchronous pulse generator and weekly timer.
* Develop control routines for time sequence process.

**Module Contents:**

|  |  |  |
| --- | --- | --- |
|  | **Topic** | **Page No.** |
| 3.1 | Introduction | 3 |
| 3.2 | ON/OFF delay timer | 4 |
| 3.3 | Asynchronous pulse generator | 6 |
| 3.4 | Weekly timer | 8 |
| 3.5 | Practical tasks | 9 |
| 3.6 | Module Exercises | 15 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **3.1 Introduction**  More complex systems cannot be controlled with logic programming blocks alone. The main reason for this is that we cannot, or choose not to add sensors to detect all conditions. In these cases we can use events to estimate the condition of the system as we did in the previous module or simply we can use timers. Typical events used by a PLC include;   * Time since an input turned on/off. * Count of occurred events.   The common theme for the mentioned events is that they are based upon one of two questions "How many?" or "How long?"  In PLC Fundamentals course you studied and used two types of timers:  **ON-delay timer**: The output of this timer is not switched on until a configured delay time has expired.  **OFF-delay timer**: The output of this timer stays high until a defined time has expired.   |  | | --- | | untitled.bmp | | Figure 3.1 ON-delay timer and OFF-delay timer |   In this module you will learn how to use the previous timers in time sequence processes in addition to the following timers:   * ON/OFF delay timer * Asynchronous pulse generator * Weekly timer   Timers are very important for industrial applications as well as for both residential and commercial applications, Some examples of their use include:   * Flashing light control and traffic light signals * Motor soft-start delay control: Instead of starting large electric motors by switching full power from a dead stop condition, reduced voltage can be switched for a "softer" start. * Conveyor belt sequence delay: especially when multiple conveyor belts are arranged to transport material * Subroutines that take place consecutively for specific time.   **3.2 ON/OFF delay timer**  The ON/OFF delay timer can be thought as a combination of two timers ON delay timer and OFF delay timer, basically it is used to set an output upon expiration of an ON time and then reset it again upon expiration of an OFF time.  Figure 3.2 shows an example of timing diagram for ON/OFF delay timer, since the ON delay time is 2 seconds for the timer, the output Q goes ON 2 seconds after switching ON I1 and it remains ON 3 seconds after switching OFF I1 because the OFF delay is 3 seconds. If the switching ON time is less than the ON delay time the output stays OFF.   |  | | --- | | untitled.bmp | | Figure 3.2 ON/OFF delay Timer |   A simple control task that needs an ON/OFF delay timer can be as follows:  A normally open PB and a normally closed PB are used to turn a motor ON and OFF respectively. The motor starts rotating 10 seconds after the ON-PB is pressed, and it stops 15 seconds after the OFF-PB is pressed.   |  |  | | --- | --- | | **Inputs** | | | Input | Assigned address | | PB1 (N.O) | I1 | | PB2 (N.C) | I2 | | **Output/s** | | | Output | Assigned address | | M1 | Q1 |   Table 3.1 I/O assignment  Latching relay is required in this example because PBs are used, the solution is shown in figure 3.3   |  | | --- | | untitled.bmp | | Figure 3.3 Motor control using ON/OFF delay timer |   **Conduct Lab activity (1) and Lab activity (2)**  **3.3 Asynchronous pulse generator**  A pulse wave or **pulse train** is a [waveform](http://en.wikipedia.org/wiki/Waveform) that has a rectangular shape. Pulse train is similar to a [square wave](http://en.wikipedia.org/wiki/Square_wave), but does not have the symmetrical shape associated with a perfect square wave.  ON time is not necessarily equal to OFF time of the wave.   |  | | --- | | untitled.bmp | | Figure 3.4 Pulse train |   Asynchronous pulse generator programming block is used to generate pulse train; the programmer can adjust both the ON time and the OFF time.  This programming block is very helpful in both time sequence processes and loop processes; a simple example is a traffic light signal.   |  | | --- | | untitled1.bmp | | Figure 3.5 Asynchronous pulse generator |   Asynchronous pulse generator has two inputs (En & INV). **En** input is used to enable the pulse train at the output, when En is high the output goes high for TH which is the pulse width then the output goes low for TL which is the Inter-pulse width. The output remains at logic low as long as En input is low. **INV** input can be used to invert the output signal. The input block INV only inverts the output signal if the block is enabled.  Example: A selector switch (I4) is used to blink the green light (Q1) in the Edutrainer kit; the light goes ON for 3 seconds and it goes OFF for 4 seconds. This loop will continue until the system is turned off using the same selector switch. Draw the FBD for this system.   |  | | --- | | untitled2.bmp | | Figure 3.6 Blinking light |   Change your program in the previous example so that the white light goes ON when the green light is OFF and white light turns OFF when the green light is ON. AND block is used to make sure that both outputs are initially OFF   |  | | --- | | kam sign.bmp | | Figure 3.7 Two blinking light bulbs |   **Conduct Lab activity (3)** |

|  |  |  |
| --- | --- | --- |
| **3.4 Weekly timer**  In many applications there is a need to use real time clock routines, like school bell and periodic lubrication process for machines. Weekly timer can be used to switch ON and switch OFF an output at certain times during the week days. Each weekly timer is equipped with three options (cams). You can configure the ON and OFF time for each individual cam.  A conflict is generated in the weekly timer when the set on time and the set off time at another cam are identical. In this case, cam 3 takes priority over cam 2, while cam 2 takes priority over cam 1.  Example: in a certain machine, lubrication pump (Q4) should work every day from 8:00 am till 8:30 am except on Fridays it shouldn’t work. On Tuesdays and Thursdays the pump should work one more time from 1:00 pm till 1:30 pm. Draw the FBD for this task.   |  | | --- | | untitled2.bmp | | Figure 3.8 Weekly timer |   Note: LOGO! must be equipped with an internal real-time clock if weekly timer is to be used.  **Conduct Lab activity (4)** |

|  |
| --- |
| **3.5 Practical tasks**  **Lab activity 1: ON/OFF delay timer** |
| Objective: understand and use ON/OFF delay timer   1. Try the following control routines, for how many seconds the green light stays ON in each case? write down your comments:  |  |  | | --- | --- | | untitled.bmp | | | **Action** | **Comments** | | Press the green PB for three seconds and then release it. |  | | Press the green PB for seven seconds and then release it. |  | | untitled2.bmp | | | **Action** | **Comments** | | Switch ON the selector switch for 10 seconds then switch it OFF |  |  1. Which input is better to be used with the ON/OFF programming block switch or pushbutton? Why? 2. The conveyor belt (Q8) is required to start 10 seconds after switching ON the selector switch (I4), and it must stay ON for 20 seconds after switching OFF the same selector switch. 3. Create and test the program for this task using one timer only.  |  | | --- | |  |  1. Create and test the program for this task using two timers.  |  | | --- | |  |  1. Compare between program (a) and program (b) in terms of the advantage for each one? |
| **Lab activity 2: Two-routine time sequence process** | | |
| **Objective:** use single timer and multiple timers with memory flag to create a two-routine time sequence process.  Use the Edutrainer kit and LOGO! soft comfort software to create and implement the following control task:  The system is started with a selector switch (I4), once the switch is turned ON the conveyer belt (Q8) will start moving for 10 seconds. After 10 seconds the belt will turn OFF for 5 seconds, as soon as the belt stops moving a green light (Q1) turns ON to indicate that the bottle is being filled. This loop will run for infinite time unless the system is turned OFF using the main selector switch.   1. Draw the FBD for this task.   (Note: there are many correct solutions for this task)   |  | | --- | |  |  1. Run and test your program. | | |
| **Lab activity 3: Asynchronous pulse generator** | |
| **Objective:** use asynchronous pulse generator programming block in time sequence processes.   1. Repeat lab activity 2 using Asynchronous pulse generator block 2. Draw the FBD for the previous control task.  |  | | --- | |  |  1. Run and test your program.  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 1. Table 3.2 shows the ON time for each light in a traffic light signal. Create the function block diagram that can be used to control this traffic light signal as follows: | |  |  | | --- | --- | | **Light** | **ON time** | | Red (Q1) | 30 Sec | | Yellow (Q2) | 5 Sec | | Green (Q3) | 20 Sec |   Table 3.2 Traffic signal | | Red🡪Yellow🡪Green🡪 Red (again) | |   Hint: See the timing diagram given below. Use Asynchronous pulse generator with the red light. (there is more than one solution)   |  | | --- | | untitled2.bmp | | Timing diagram for traffic light signal | | |

|  |
| --- |
| Solution |

|  |  |
| --- | --- |
| **Lab activity 4: Weekly timer** | |
| **Objective:** use Weekly timer programming block.   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | You are required to create a program that can be used to control ATHS bell (Q1), the bell should ring for 1 minuet at the end of each period, and at the end of each break, timings are given in the table. | |  |  | | --- | --- | | **Period** | **End time** | | 1 | 8:15 am | | 2 | 9:05 am | | 3 | 9:55 am | | Break | 10:20 am | | 4 | 11:05 am | | 5 | 11:55 am | | 6 | 12:45 pm | | 7 | 1:35 pm | | Break | 2:15 pm | | 8 | 3:00 pm | |  1. Draw the FBD for this task.  |  | | --- | |  |  1. Use the LOGO! Soft comfort to simulate and test your program (you can adjust the day and time for testing). | |
| **3.6 Module Exercises** | |
| |  |  |  | | --- | --- | --- | | 1. A pneumatic cylinder is used in a process. The cylinder can become stuck, and we need to detect this. Limit switches are added to both endpoints of the cylinder’s travel to indicate when it has reached the end of motion (I1, I2). If the cylinder takes more than 3 seconds to complete a motion this will indicate a problem. When this occurs a light (Q1) turns on. Develop FBD that will watch for this failure. | untitled1.bmp | | | 1. Every school day the National anthem is to be played at 7:30 am (Q1) and the school bell should be activated at 3:00 pm (Q2). Use the weekly timer programming block to create FBD for this task. | untitled3.bmp | | | 1. A tank is to be filled with 2 chemicals, when PB1(N.O)🡪 I1 is pressed pump1🡪Q1 starts. After 10 seconds the proper amount of chemical 1 has been pumped, pump 1 shuts OFF and pump2🡪Q2 runs for 5 seconds adding chemical 2. Then pump 2 stops and a mixer🡪Q3 works for 2 minuets. PB2 (N.C)🡪 (I2) can be used to terminate the process at any time. Develop the program that does this task. | | | untitled.bmp | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | 1. In dangerous processes it is common to use two pushbuttons that require an operator to use both hands to start a process (this keeps hands out of presses, etc.). To develop this there are two inputs (I1 and I2) that must be turned on within 0.25s of each other before a machine cycle (Q1) may begin. Create the program for this task. | untitled3.bmp | | 1. Create a function block diagram for a traffic light signal that has the following sequence:   Red🡪Yellow🡪Green🡪 Green blinks 3 times 🡪Yellow🡪 Red (again)  35sec🡪3sec 🡪25sec🡪 1sec ON-1sec OFF 🡪 3sec 🡪 35sec | | | untitled3.bmp | | |